



**UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA**

Paradigmas de Programación

Laboratorio N°3

Alumno: Cristhofer Parada Salazar

Sección: B-2.

Profesor: Víctor Flores Sánchez.

Santiago – Chile
1-2021

TABLA DE CONTENIDOS

Contents

Alumno: Cristhofer Parada SalazarSección: B-2	2
1.1.1 Descripción del problema.....	4
1.1.2 Descripción del paradigma utilizado	4
2.1.1 Análisis del problema respecto a sus requisitos específicos.....	5
Para el desarrollo de la solución se solicitan los siguientes requisitos funcionales específicos:	5
2.1.2 Diseño de la solución.	6
2.1.3 Aspectos de implementación.....	10
2.1.4 Instrucciones de uso.	11
2.2 Resultados y autoevaluación.	12
2.3 Conclusión.	14
CAPÍTULO 3. ANEXO	15

CAPÍTULO 1. INTRODUCCIÓN

El siguiente informe tiene como objetivo mostrar al lector todos los detalles relacionados al Laboratorio 3 de Paradigmas de Programación, los cuales a grandes rasgos pueden ser resumidos como descripción del problema, descripción del paradigma utilizado, el cómo se resolvió el problema y las diversas estructuras, TDAs o clases que se utilizaron para la implementación de la solución, además de la forma de utilizar este programa a través de Javac y la explicación de las diversas funciones implementadas.

1.1.1 Descripción del problema

En el semestre 1/2021, a los alumnos de la asignatura de Paradigmas de Programación, se les solicitó la implementación de una Red Social, la cual tenía como funciones principales acciones como registrarse en la red social, crear publicaciones, comentar publicaciones, seguir a otros usuarios, etc. Para ello, se les solicitó a los alumnos que el programa debiese estar escrito en el lenguaje de Java y esto conlleva a seguir el patrón MVC (model, visual, control), es decir, se distribuirá el código en distintas secciones, lo cual permite poder clasificar la información, la lógica del sistema y la interfaz que se le presenta al usuario.

1.1.2 Descripción del paradigma utilizado

En esta ocasión, se solicitó que se utilice el paradigma POO (Programación Orientada a Objetos) para crear la solución al problema. POO tiene características bastante representativas, como la abstracción donde buscamos los detalles fundamentales que puede tener un objeto, el polimorfismo y la herencia, donde un objeto adquiere las propiedades y métodos de otro objeto. En POO encontramos clases, que son modelos que definen atributos y métodos comunes, a todos los objetos que sean de dicha clase.

CAPÍTULO 2. SOLUCIÓN DEL PROBLEMA

2.1.1 Análisis del problema respecto a sus requisitos específicos.

Para el desarrollo de la solución se solicitan los siguientes requisitos funcionales específicos:

- Implementar abstracciones apropiadas para el problema mediante el uso de clases y estructuras que modelen de forma correcta el problema entregado.
- Función authentication: Función que permite que registrar usuarios en una red social, iniciar sesión en una red social o desconectarse de una red social.
- Función post: Permite a un usuario con sesión iniciada realizar una nueva publicación, la cual puede ser propia o con usuarios etiquetados
- Función follow: Permite al usuario con sesión iniciada poder seguir a otro usuario.
- Función share: Funcionalidad que permite al usuario con sesión iniciada compartir el contenido de una publicación.
- Función visualize: permite crear un string y mostrarlo por pantalla con toda la información que tiene la red social, en caso de que no haya ningún usuario conectado, o la información específica de dicho usuario en caso de que haya un usuario en línea.

Además de las funciones antes mencionadas, se da la libertad de escoger otras funciones que se pueden implementar en el programa, específicamente en este caso, se implementaron las funciones de comment, like y viral:

- Función comment: permite a un usuario comentar una publicación siempre y cuando esta última exista.
- Función react: permite a un usuario con sesión iniciada agregar una reacción a una publicación, siempre y cuando esta última exista.
- Función viral: permite consultar dada una lista de publicaciones, cuales tienen un mínimo de K compartidas.

2.1.2 Diseño de la solución.

El diseño de esta solución consiste en crear distintas clases que representen de manera apropiada el problema, con eso en mente, se implementaron las siguientes clases que se presentarán a continuación.

Clase User:

Corresponde a la presentación de un usuario en una red social. La clase User tiene como atributos una id, única para cada usuario, un nombre de usuario como string, una contraseña para autenticar al usuario, una fecha de creación de la cuenta, una lista de Users que corresponden a los followers del usuario, una lista de Post que corresponden a las publicaciones hechas por el usuario. Además de tener como métodos addFollower() y addUserPost().

Clase Post:

Clase correspondiente a lo que sería una publicación dentro de la red social. La clase Post tiene como atributos una id única, un String que corresponde al tipo de publicación (ya sea texto, imagen, audio), un String con el contenido de la publicación, una fecha de creación del post, una lista de tipo Reacts con las reacciones que ha recibido la publicación, una lista del tipo Post que corresponde a los comentarios, una lista de Strings con los etiquetados de la publicación y una lista de Users correspondiente a los usuarios que han compartido la publicación. Además, tiene como métodos addReact() para agregar una reaccion, addTag() para agregar un etiquetado y addShared() para agregar a un usuario que haya compartido la publicación.

Clase React:

Clase correspondiente a las reacciones que se le puede agregar a una publicación. La clase React tiene como atributos una ID única, un autor de la reacción, una fecha de creación de la reacción, el contenido de la reacción y el tipo de reacción (puede ser angry, haha, like, sad, love).

Clase SocialNetwork:

Clase principal del programa, en donde interactúan y se agrupan las demás clases. Tiene como atributos un String representando el nombre de la red social, una fecha de creación de la red social, una lista de tipo User con los usuarios registrados en la red social, una lista de tipo Post con las publicaciones y comentarios realizados en la red social, una lista tipo Reacts con las reacciones que se han realizado en la red social, un atributo tipo User con toda la información del usuario online y una variable tipo boolean que marca si hay un usuario online o no.

```

### Bienvenido a LMAOBOOK ###
Escoja la opcion que desea realizar:
1. LOGEARSE
2. REGISTRARSE
3. VER RED SOCIAL
4. IS VIRAL?
5. SALIR
Introduzca su eleccion:
3
Su opcion fue la numero 3: Ver el contenido de la red social
Nombre de la red social: LMAOBOOK
Fecha de creacion de la red social: 12/08/2021
*-----Usuarios de la red social-----*

Nombre de usuario: Katsugo
Cuenta creada con la fecha de: 12/08/2021
El usuario Katsugo cuenta con 1 followers, los cuales son:
    Harunomi
El usuario cuenta con 3 publicaciones, las cuales son:
#### Publicacion numero1 ####
Publicacion creada con fecha: 12/08/2021
Tipo de publicacion: audio
Contenido de la publicacion: cover.mp3
La publicacion cuenta con 0 reacciones
La publicacion actual tiene 0 comentarios

La publicacion tiene como etiquetados a: La publicacion ha sido compartida 0 veces, por:

#### Publicacion numero2 ####
Publicacion creada con fecha: 12/08/2021
Tipo de publicacion: audio
Contenido de la publicacion: song.mp3
La publicacion cuenta con 0 reacciones
La publicacion actual tiene 0 comentarios

```

Figura 1: menú inicial del programa, mostrando algunas opciones y la red social precargada con datos.

Con dicho menú creado, tenemos distintas alternativas para manejarnos dentro de la red social, principalmente se sugiere poder iniciar sesión como “Harunomi” con contraseña “password1”, luego de eso se mostrará un menú secundario con las demás opciones disponibles para el usuario:

```

Su opcion fue la numero 1: logearse en la red social
Ingrese el nombre de usuario:
Harunomi
Ingrese la contrasena para el usuario Harunomi
password1
****Bienvenido a LMAOBOOK****
Escoje una accion a realizar
1. Realizar una publicacion
2. Seguir a un usuario
3. Compartir una publicacion
4. Ver su perfil
5. Comentar una publicacion
6. Agregar una reaccion a una publicacion
7. Busca dada una lista de IDs que publicaciones han sido compartida K veces
8. Cerrar sesion en la red social
Ingrese una de las opciones anteriores:

```

Figura 2: Segundo menú disponible para un usuario con sesión iniciada.

Para llevar a cabo la implementación de la solución se hizo uso de diagramas UML para poder conceptualizar de mejor manera la forma en la que se relacionan las clases:

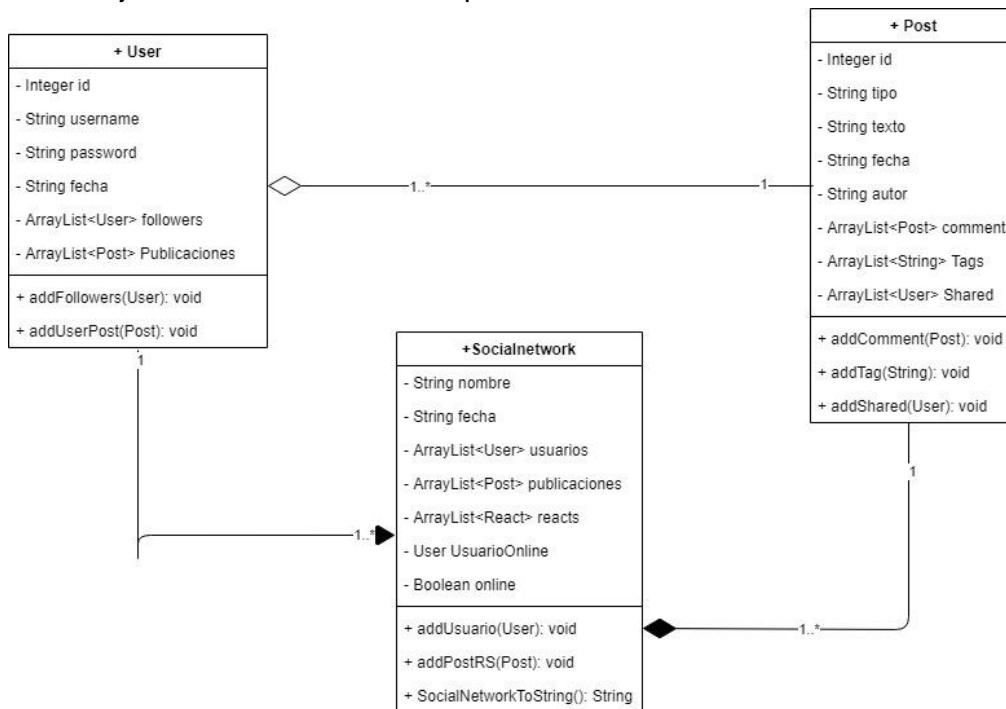


Figura 3: Diagrama de clases al comienzo del proyecto

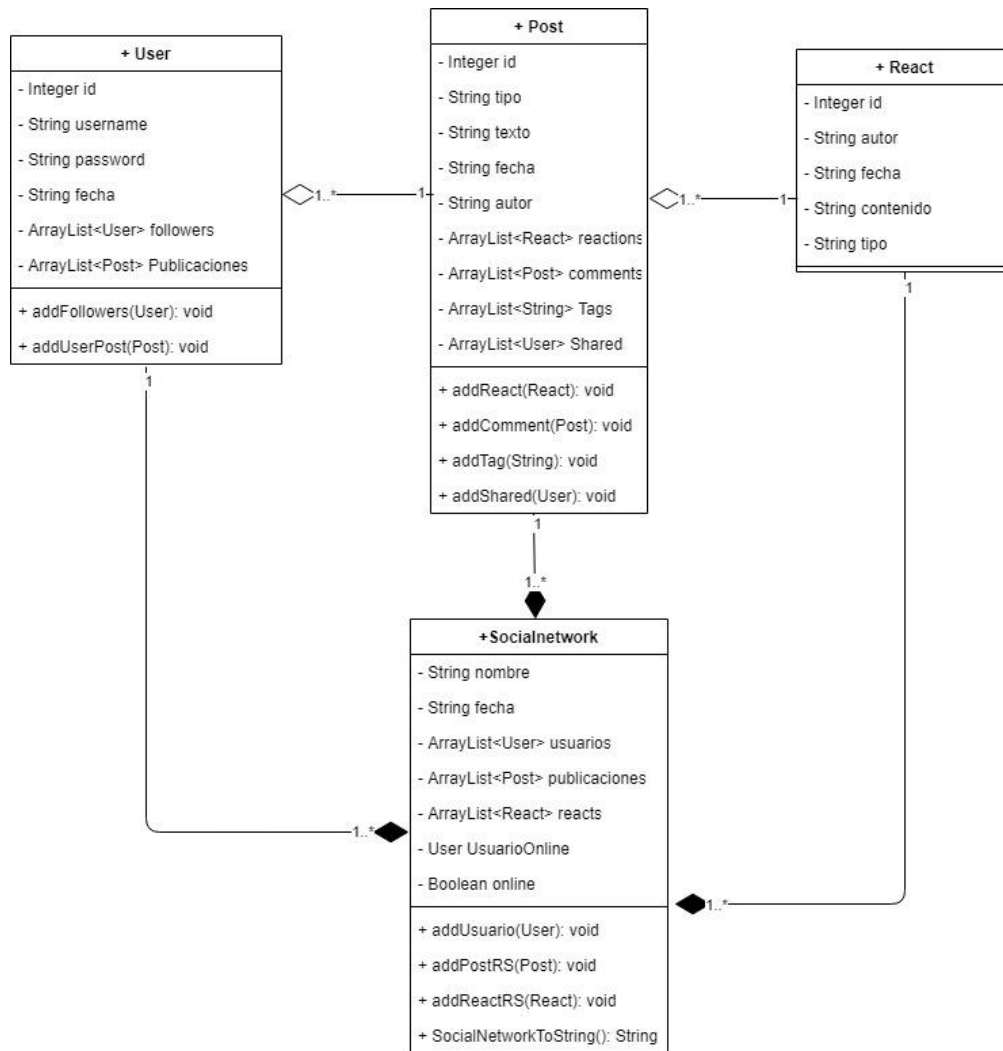


Figura 4: Diagrama UML de la solución final

Como se puede ver, en este caso, a la solución final se le agregó una nueva clase, la clase React, la cual busca modelar de mejor manera las reacciones que se le dan a las publicaciones, además abre las posibilidades de tener más de un tipo de reacciones, además de la opción de dejar una especie de “comentario extra” a la reacción.

2.1.3 Aspectos de implementación.

Este programa fue escrito en el lenguaje de Java y el IDE utilizado fue Microsoft Visual Studio, apoyado por la extensión “Java Extension Pack”. Se hizo uso de librerías tales como `java.util.Date`, `java.util.ArrayList`, `java.text.SimpleDateFormat`, `java.util.Scanner`, `java.util.InputMismatchException`, las cuales son librerías estándares de java.

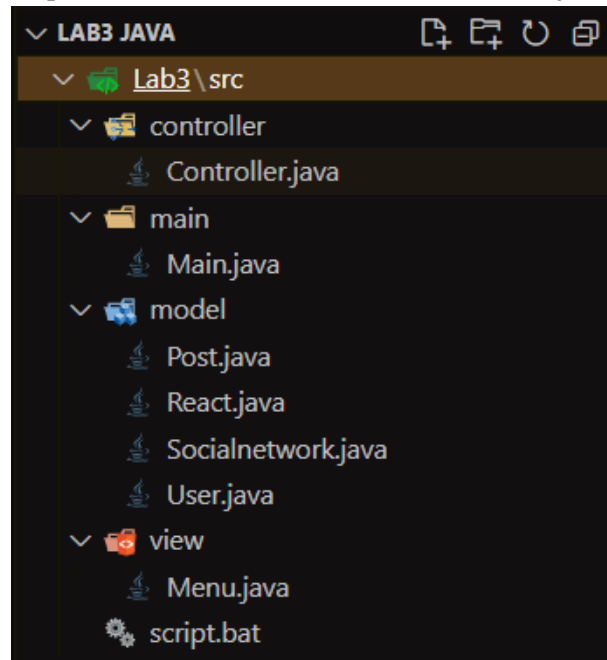


Figura 5: Organización del programa

El programa este compuesto por 3 carpetas, las cuales corresponden a las carpetas siguiendo el formato MVC.

En la carpeta model, tenemos 4 archivos los cuales corresponden a las clases que se pueden ver en la figura 4, en esta carpeta encontramos los archivos “Post.java”, “React.java”, “Socialnetwork.java” y “User.java”.

En la carpeta view podemos encontrar el archivo “Menu.java” el cual esta a cargo de todo lo visual que podemos ver en la consola y controlar las acciones que el usuario es capaz de realizar.

En la carpeta controller podemos encontrar el archivo “Controller.java”, clase que permite que el usuario realice todas las acciones que requiera por parte del menú, en esta clase podemos encontrar las funciones que se han requerido

En la carpeta main podemos encontrar el archivo main.java, el cual crea una red social, crea un controlador, crea un menú y hace el llamado final al menú.

2.1.4 Instrucciones de uso.

En primer lugar, verificar que se tenga instalado Java 11, y en la carpeta donde se encuentre el src ejecutar abrir una consola en esa dirección. Con ello, podemos ejecutar el “script.bat” el cual nos compilará todos los archivos y nos llevará directamente al menú, una vez dentro del programa es cosa de seguir las instrucciones que se nos muestre por pantalla para poder interactuar con el programa.

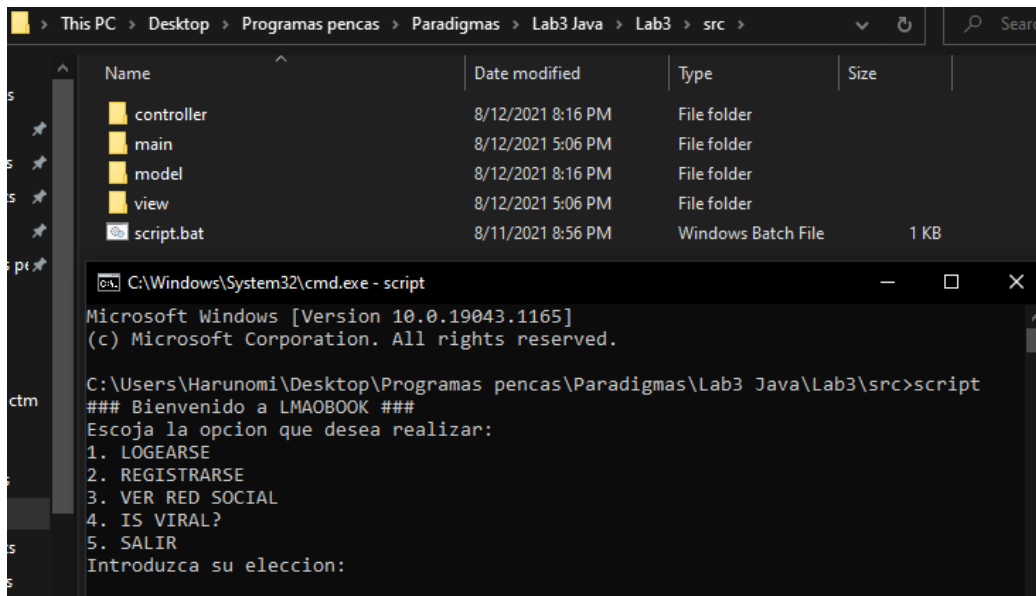


Figura 6: ejecución del programa

2.2 Resultados y autoevaluación.

En esta sección se hará una revisión de los resultados, para ello se verificarán los requisitos no funcionales y funcionales solicitados

Requerimientos no funcionales:

Lenguaje Java obligatorio	Logrado: el lenguaje usado en el programa es Java.
Versión: utiliza OpenJDK en su versión 11	Logrado: la versión del JDK es la 11.05
Interacciones con el programa: Todas las interacciones con el programa deben ser mediante consola/terminal.	Logrado: El usuario solamente puede interactuar por la red social mediante el uso del menú por consola.
Uso del paradigma	Logrado: Se implemento de manera correcta el paradigma orientado a objetos
Organización del código	Logrado: El programa se encuentra bien organizado siguiendo el patrón MVC.
Diagrama de análisis	Logrado: Se crea un diagrama de análisis previo a la implementación de la solución.
Diagrama de diseño	Logrado: Se presenta un diagrama final con la verdadera implementación de la solución donde se muestra las clases finales.
Historial: al menos 10 commits en un mínimo de 2 semanas, con un mínimo de 10 commits.	No logrado: Si bien es cierto se logró el mínimo de commits, estos no logran el rango de ser entre 2 semanas. Primer commit: 04/08/2021. ultimo commit: 12/08/2021
Prerequisitos de funciones:	Logrado, cada función tiene su prerequisite implementado como se solicitó.

Requerimientos Funcionales:

Todas las funciones cumplen con los requerimientos funcionales exigidos por la coordinación, por lo que este punto también fue respetado en su gran totalidad.

2.3 Conclusión.

Concluyendo, finalmente, se puede concluir que Java es un lenguaje de programación bastante poderoso, se siente bastante cómodo para escribir y además mantener un orden entre distintos tipos de clases. Se puede sentir como tanto el programador como el usuario puede tener control sobre ciertos objetos.

En este caso, no se logró cumplir con el rango de commits, por temas de carga académica, pero se está bastante satisfecho con el trabajo logrado, hubo un buen aprendizaje del paradigma y se pudo crear una implementación al problema acorde a lo que se requería. Aspectos a mejorar, sigue siendo intentar mantener una buena organización personal, para poder mantener la misma cantidad de commits, pero en el rango que se pide de 2 semanas.

CAPÍTULO 3. ANEXO

```
Nombre de usuario: Victor Flores
Cuenta creada con la fecha de: 12/08/2021
El usuario Victor Flores cuenta con 0 followers, los cuales son:
El usuario cuenta con 0 publicaciones, las cuales son:

Nombre de usuario: Rose isa
Cuenta creada con la fecha de: 12/08/2021
El usuario Rose isa cuenta con 0 followers, los cuales son:
El usuario cuenta con 0 publicaciones, las cuales son:*
```

Anexo 1: registro adicional de usuarios

```
Ingrese el nombre de usuario:
Harunomi
Ingrese la contraseña para el usuario Harunomi
password1
****Bienvenido a LMAOBOOK*****
Escoja una accion a realizar
1. Realizar una publicacion
2. Seguir a un usuario
3. Compartir una publicacion
4. Ver su perfil
5. Comentar una publicacion
6. Agregar una reaccion a una publicacion
7. Busca dada una lista de IDs que publicaciones han sido compartida K veces
8. Cerrar sesion en la red social
Ingrese una de las opciones anteriores:
3
Ingrese la ID de la publicacion que desea compartir
7
Ingrese la cantidad de etiquetados que desea que tenga la publicacion compartida:
1
Cantidad de etiquetados seleccionada: 1
Ingrese el etiquetado numero 1
Katsugo
Publicacion compartida con exito!
```

Anexo 2: Usuario Harunomi compartiendo la publicación 7 etiquetando a Katsugo.

```
****Bienvenido a LMAOBOOK****
Escoje una accion a realizar
1. Realizar una publicacion
2. Seguir a un usuario
3. Compartir una publicacion
4. Ver su perfil
5. Comentar una publicacion
6. Agregar una reaccion a una publicacion
7. Busca dada una lista de IDs que publicaciones han sido compartida K veces
8. Cerrar sesion en la red social
Ingrese una de las opciones anteriores:
6
Ingrese la ID de la publicacion o comentario la cual desea agregar una reaccion
4
Ingrese el tipo de reaccion (puede ser del tipo: like, haha, angry, sad, love)
love
Ingrese el texto que desea que lleve la reaccion:
horrible
Reaccion realizada con exito!
```

Anexo 3. Usuario agregando una reacción a una publicación.